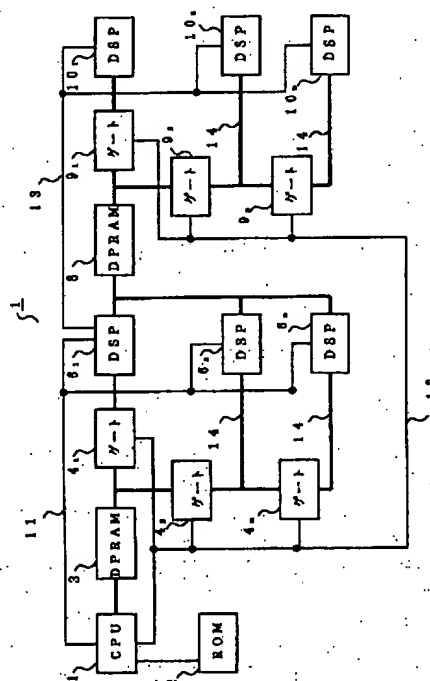


(11)特許出願公開番号

(43)公開日 平成11年(1999)8月10日

420K

(74) 代理人 弁理士 井上 俊夫



【特許請求の範囲】

【請求項1】システム起動時に読み込まれるべきプログラム情報を格納するプログラム格納部と、前記プログラム格納部に格納されたプログラム情報を入力することによってブートする複数のプロセッサ部と、前記プログラム格納手段と前記プロセッサ部との間にそれぞれ介在し、各プロセッサ部に入力されるべきプログラム情報の伝送経路を入状態または断状態とする複数のゲート部と、ブート対象となるプロセッサ部に対応するゲート部を入状態とするとともに、他のゲート部を断状態とする状態制御部と、を備えることを特徴とするプログラムブート回路。

【請求項2】システム起動時に読み込まれるべきプログラム情報を格納するプログラム格納部と、前記プログラム格納部に格納されたプログラム情報を入力することによってブートする複数のプロセッサ部と、プロセッサ部によって前記プログラム格納手段格納されたプログラム情報を読み出すタイミングを調整するタイミング調整部と、ブート対象となるプロセッサ部に対し、他のプロセッサ部における読み出しタイミングを所定時間遅らせるタイミング制御部と、を備えることを特徴とするプログラムブート回路。

【請求項3】前記プログラム格納部の後段に、単一のバスで接続され、同時に読み書き可能なメモリ部を設けることを特徴とする請求項1または2記載のプログラムブート回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、移動通信システムの基地局内の無線送受信装置等におけるプログラムブート回路に関する。

【0002】

【従来の技術】プログラムブートとは、自身内にプログラムを格納しておくための記憶部（たとえば、ROM: Read Only Memory）を持たないCPU（Central Processing Unit）やDSP（Digital Signal Processor）に対して外部からプログラム情報を与えることをいう。

【0003】従来、このようなプログラムブート回路としては、たとえば、特開平8-263297号公報に示すようなプログラムブート方式がある。

【0004】図3は、従来のプログラムブート回路の回路構成を示すブロック図である。同図において、プログラムブート回路100は、ブートを管理するCPU101と、ブートプログラムを格納するROM102と、データを仲介するDPRAM（Dual Port Random Access Memory）1031～103n、1051～105nと、ブートする必要のあるDSP1041～104n、1061～106nとを備えている。そして、CPU101

と各DSP1041～104n、1061～106nとはリセット線111によって接続され、また、CPU101、DPRAM1031～103n、1051～105n、DSP1041～104n、1061～106nの各間はバス112によって接続されている。

【0005】以上の構成において、DSP1041～104nをブートするためには、まず、CPU101がROM102からプログラムを読み込み、読み込んだプログラムをバス112を介してDPRAM1031～103nに転送する。そして、CPU101からリセット線111を介してDSP1041～104nをリセット解除を行うことにより、DSP1041～104nはDPRAM1031～103nからプログラムを読み込んでブートする。

【0006】ここで、ブートすべきDSPを増やす場合、最も単純な方法は、CPU101をすべてのDPRAM1031～103n、1051～105nとバス111で接続することであるが、この方法では増やしたDSPの数に比例して回路構成が複雑になってしまう。このため、図3に示す従来例では、DPRAMをDPRAM1031～103nと、DPRAM1051～105nとに分け、CPU101は、DPRAM1031～103nとの間だけバス112で接続するとともに、DSP1041～104nのリセット解除を行う。

【0007】そして、DSP1041～104nは、DPRAM1051～105nとの間をバス112で接続するとともに、DSP1061～106nのリセット解除を行う。すなわち、下位に位置するDSP1061～106nに対しては、上位に位置するDSP1041～104nによってブート制御することにより、回路構成を複雑化することなくDSPの増加を図ることができる。

【0008】また、DPRAM1031～103n、1051～105nのメモリ空間に、転送要求フラグ、行き先種別エリア、起動報告エリア等を設け、これらをCPU101によって監視することで、すべてのDSP1041～104n、1061～106nのブートを行っていた。

【0009】

【発明が解決しようとする課題】しかしながら、このような従来のプログラムブート回路にあっては、CPU101とDPRAM1051～105nとの間を直接接続するためのバスを設ける必要がないので、DSPを増加させる場合の回路構成の簡略化は図れるものの、CPU101とDPRAM1031～103nとの間のバス112のさらなる減少という要求には応えられないという問題点があった。

【0010】また、CPU101とDSP1041～104n、1061～106nの間にはそれぞれDPRAM1031～103n、1051～105nが介装さ

れているため、たとえば、CPU101からみて上位に位置するDSP1041～104nのいずれかが故障した場合、下位に位置するDSP1061～106nはプログラムをブートできないおそれが生じる。

【0011】本発明の課題は、上記問題点を解決するためになされたものであり、バスラインを低減するとともに、確実なプログラムブートを行うことのできるプログラムブート回路を提供することにある。

【0012】

【課題を解決するための手段】本発明のプログラムブート回路は、システム起動時に読み込まれるべきプログラム情報を格納するプログラム格納部と、プログラム格納部に格納されたプログラム情報を入力することによってブートする複数のプロセッサ部と、プログラム格納手段とプロセッサ部との間にそれぞれ介在し、各プロセッサ部に入力されるべきプログラム情報の伝送経路を入状態または断状態とする複数のゲート部と、ブート対象となるプロセッサ部に対応するゲート部を入状態とするとともに、他のゲート部を断状態とする状態制御部とを備えるように構成している。

【0013】また、システム起動時に読み込まれるべきプログラム情報を格納するプログラム格納部と、プログラム格納部に格納されたプログラム情報を入力することによってブートする複数のプロセッサ部と、プロセッサ部によってプログラム格納手段格納されたプログラム情報を読み出すタイミングを調整するタイミング調整部と、ブート対象となるプロセッサ部に対し、他のプロセッサ部における読み出しタイミングを所定時間遅らせるタイミング制御部とを備えるように構成している。

【0014】この場合、プログラム格納部の後段に、単一のバスで接続され、同時に読み書き可能なメモリ部を設けることが好ましい。

【0015】

【発明の実施の形態】以下、図示した一実施形態に基づいて本発明を詳細に説明する。

【0016】図1は、本発明の一実施形態におけるプログラムブート回路の回路構成を示すブロック図である。同図において、プログラムブート回路は、ブート制御を統括管理するCPU（状態制御部）1と、ブートプログラムを格納するROM（プログラム格納部）2と、データを仲介するDPRAM（メモリ部）3、8と、ゲート（ゲート部）41～4n、91～9nと、ブート対象となるDSP（プロセッサ部）61～6n、101～10nとを備えている。

【0017】そして、CPU1と各DSP61～6nとはリセット線11によって接続され、また、CPU1と各ゲート41～4n、91～9nとは制御線12によって接続されている。さらに、DSP61と各DSP101～10nとはリセット線13によって接続されている。なお、CPU1、DPRAM3、ゲート41～4

n、DSP61～6n、DPRAM8、ゲート91～9n、DSP101～10nの各間はバス14によって接続されている。

【0018】次に、上述の実施形態におけるプログラムブート回路の動作例を図1に基づいて説明する。

【0019】まず、プログラムブート回路に通電すると、CPU1はDPRAM3、8のメモリクリア等の初期化処理を実行し、続いてCPU1はROM2内に格納されているブートプログラムを読み出す。ブートプログラムの読み出しが終了すると、CPU1はブートプログラムをDPRAM3に転送する。

【0020】ここで、たとえば、DSP61を最初にブートする場合、CPU1はDPRAM3におけるメモリ空間の転送先アドレスにDSP61のフラグを立て、ゲート42～4nの入出力を分離するように制御線12をコントロールする。そして、ゲート41だけを入出力が可能な状態とし、DPRAM3とDSP61とだけが接続するようにする。

【0021】そして、CPU1はリセット線11、13のうち、DSP61に接続されたリセット線11だけを解除し、DSP61はDPRAM3の転送先アドレスを参照することにより自分がブートを行うことを確認し、DPRAM3よりプログラムをブートする。ブートが完了したDSP61はDPRAM3に対して終了報告のフラグを立てる。CPU1はDPRAM3の終了確認のフラグを監視して次のDSPのブートを開始する。

【0022】次いで、DSP62をブートする場合、CPU1はDPRAM3の転送先アドレスをDSP62のものに書き換える。その後、制御線12によってゲート41、43～4nの入出力を分離し、ゲート42だけ入出力が可能な状態とする。そして、CPU1はDSP62のリセット線11だけを解除し、DSP62はDPRAM3の転送先アドレスが自分のものであることを確認した後、ブートを開始する。ブート終了後、DPRAM3に完了報告のフラグを立てる。以降、CPU1によるゲート41～4nの制御によってDSP43～4nはプログラムをブートする。

【0023】つぎに、DSP101～10nをブートする場合、CPU1はROM2から読み出したプログラムをDPRAM3に転送し、同時に転送先アドレスにDSP101を書き込む。これを読んだDSP61は、DPRAM3に転送確認のフラグを立て、プログラムをそのままDPRAM8に転送する。このとき、DSP61が故障等でプログラムを転送できない場合にはDPRAM3に対して転送確認フラグを立てることもできないため、CPU1によってタイムアウトを設けて監視することにより、一定時間まで転送確認フラグが立たない場合には、次のDSP62に対してプログラムをDPRAM8に転送するように命令する。

【0024】こうしてDPRAM8にプログラムを転送

したことを確認した後、最初にDSP101をブートする場合、CPU1はDPRAM8の転送先フラグをDSP101のものにする。そして、CPU1（もしくはDSP61）によって制御線12をコントロールし、ゲート91以外のゲート92～9nの入出力を分離し、ゲート91とDSP101とだけが接続するようにする。

【0025】つぎに、DSP61はDSP101に接続するリセット線13だけを解除することでDSP101はDPRAM8よりプログラムをブートする。ブートを終了したDSP101は、終了報告のフラグをDPRAM8に書き込み、それを確認したDSP61はDPRAM3に対して同様の終了報告のフラグを立てる。CPU1はそれを確認して次のDSPのブートを開始する、という手順で順次DSPのブートを行うことができる。

【0026】すなわち、CPUからの情報を転送する高価なDPRAMの個数を減らすとともに、その間のバス配線も省略することができるため、DSPの数量を増加することによる回路規模の増加を低減することができる。

【0027】図2は、本発明の他の実施形態におけるプログラムブート回路の回路構成を示すブロック図である。同図において、プログラムブート回路は、ブート制御を統括管理するCPU（タイミング制御部）1と、ブートプログラムを格納するROM（プログラム格納部）2と、データを仲介するDPRAM（メモリ部）3、8と、遅延回路（タイミング調整部）51～5n、71～7nと、ブート対象となるDSP（プロセッサ部）61～6n、101～10nとを備えている。

【0028】そして、CPU1と各DSP61～6nとはリセット線11によって接続され、また、CPU1と各遅延回路51～5n、71～7nとは制御線12によって接続されている。さらに、DSP61と各DSP101～10nとはリセット線13によって接続されている。なお、CPU1、DPRAM3、遅延回路51～5n、DSP61～6n、DPRAM8、遅延回路71～7n、DSP101～10nの各間はバス14によって接続されている。

【0029】次に、上述の実施形態におけるプログラムブート回路の動作例を図2に基づいて説明する。

【0030】まず、プログラムブート回路に通電すると、CPU1はDPRAM3、8のメモリクリア等の初期化処理を実行し、続いてCPU1はROM2内に格納されているブートプログラムを読み出す。ブートプログラムの読み出しが終了すると、CPU1はブートプログラムをDPRAM3に転送する。

【0031】ここで、CPU1はリセット線11を解除することで、まず最初にDSP41がDPRAM3のプログラムを読み込む。リセット線11とDSP62～6nとの間には遅延回路51～5nが介在するため、プログラムのブートに要する時間をもとに、CPU1からの

制御線12によってDSP62～6nによるプログラム読み込みタイミングを任意に設定することができる。すなわち、DSP42にリセット解除がかかかかるのは、DSP61のリセット解除から任意の時間後となり、DSP62～6nのブートは任意の時間間隔で行うことができる。

【0032】なお、DSP61に次いでDSP62をブートする際、DSP62用のプログラムがDSP61に書き込まれないように、CPU1はリセット線11の解除をするか、あらかじめDSP61のブートプログラムに、自らのブート完了後より同じ系統のDSP62、63のブートが完了するまでの間はアイドル状態を維持し、バスからの書き込みが行われないようなプログラムをあらかじめ格納しておく。

【0033】また、DSP101～10nをブートさせる場合には、DPRAM3、DSP61を経由してDPRAM8にプログラムを転送する。具体的には、CPU1はROM2から読み出したプログラムをDPRAM3に転送すると同時に転送先アドレスにDSP101を書き込む。これを読んだDSP61は、DPRAM3に転送確認のフラグを立て、プログラムをそのままDPRAM8に転送する。このとき、DSP61が故障等でプログラムを転送できない場合にはDPRAM3に対して転送確認フラグを立てることもできないため、CPU1によってタイムアウトを設けて監視することにより、一定時間まで転送確認フラグが立たない場合には、次のDSP62に対してプログラムをDPRAM8に転送するように命令する。

【0034】こうしてDPRAM8にプログラムを転送したことを確認した後、CPU1（もしくはDSP61）はリセット線13を解除することで、まず最初にDSP101がDPRAM8のプログラムを読み込む。リセット線13とDSP102～10nの間には遅延回路71～7nが介在するため、プログラムのブートに要する時間をもとに、CPU1からの制御線12によってDSP102～10nによるプログラム読み込みタイミングを任意に設定することができる。すなわち、DSP102にリセット解除がかかかかるのは、DSP61のリセット解除から任意の時間後となり、DSP102～10nのブートは任意の時間間隔で行うことができる。

【0035】以上説明したように、本実施形態では、第1の実施形態の各DSPごとに設けていたゲートを省略することができるので、各ゲートに対するタイミング制御が必要なくなり、CPUの処理工程を軽減できるというメリットを有する。

【0036】

【発明の効果】以上の説明から明らかなように、本発明によれば、外部からプログラムをブートする必要のあるDSPを備えた装置の回路構成を簡略化することで、DSPの数量を増加することを容易にすることができる。

【図面の簡単な説明】

【図1】本発明の一実施形態におけるプログラムブート回路の回路構成を示すブロック図である。

【図2】本発明の他の実施形態におけるプログラムブート回路の回路構成を示すブロック図である。

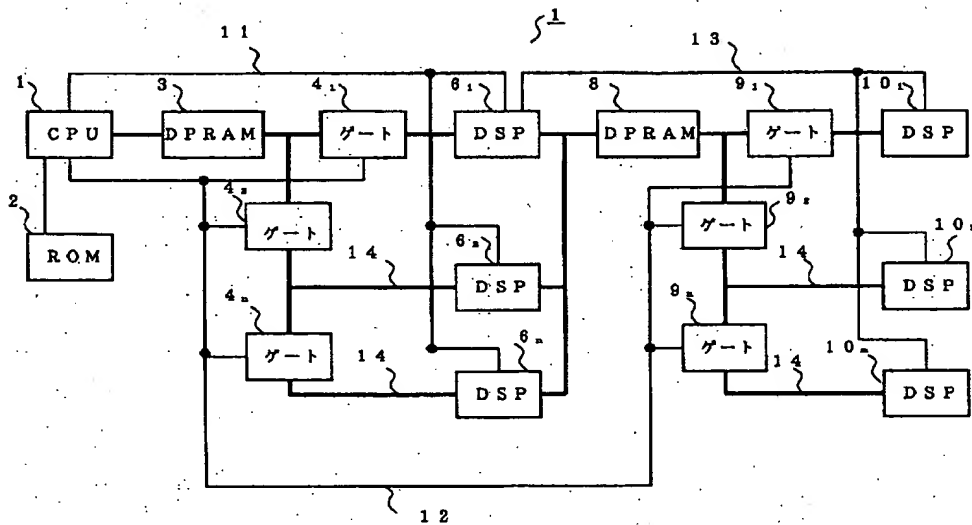
【図3】従来のプログラムブート回路の回路構成を示すブロック図である。

【符号の説明】

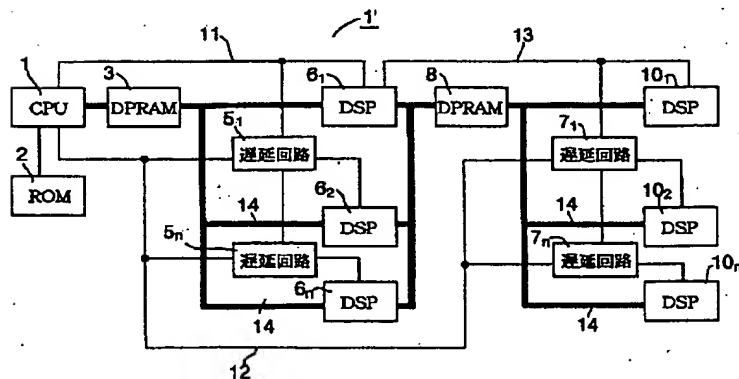
1 CPU (状態制御部)
2 ROM (プログラム格納部)
3 DPRAM (メモリ部)
41 ~ 4n ゲート (ゲート部)
51 ~ 5n 遅延回路 (タイミング調整部)
61 ~ 6n DSP (プロセッサ部)
71 ~ 7n 遅延回路 (タイミング調整部)

8 DPRAM (メモリ部)
91 ~ 9n ゲート (ゲート部)
101 ~ 10n DSP (プロセッサ部)
11 リセット線
12 制御線
14 バス
100 プログラムブート回路
101 CPU
102 ROM
1031 ~ 103n DPRAM
1041 ~ 104n DSP
1051 ~ 105n DPRAM
1061 ~ 106n DSP
111 リセット線
112 バス

【図1】



【図2】



【図3】

